

## CLAIMS

What is claimed is:

1 1. A computer system, comprising at least one processor, a system memory coupled to said  
2 processor, at least one input/output device coupled to said processor, and a watchdog timer device,  
3 wherein the computer system executes:

4 an operating system with at least two protection layers;

5 one or more key computer applications; and

6 an application watchdog driver that monitors user designated computer applications for  
7 periodic messages;

8 wherein if the watchdog driver receives a periodic message from all user-designated  
9 computer applications in a predetermined period of time, the watchdog driver delivers a command  
10 to clear the watchdog timer device.

11 2. The computer system of claim 1 further comprising:

12 a message passing interface that transmits signals between the two protection layers;

13 wherein the watchdog driver executes in one protection layer and the application executes  
14 in another protection layer and wherein the periodic message is transmitted from the application to  
15 the application watchdog driver through the message passing interface.

1 3. The computer system of claim 2 wherein:

2 the message passing interface is a shared memory queue.

1 4. The computer system of claim 1 wherein:

2 the watchdog timer device resides in a hardware layer separate from the operating system  
3 protection layers and wherein the application watchdog driver communicates with the watchdog  
4 timer device via a hardware abstraction layer.

1 5. The computer system of claim 1 further comprising a system watchdog timer device;  
2 wherein the computer system also executes a system watchdog driver that monitors the operating  
3 system for periodic messages; and  
4 wherein if the system watchdog driver receives a periodic message from the operating system in a  
5 predetermined period of time, the system watchdog driver delivers a command to clear the system  
6 watchdog timer device.

1 6. The computer system of claim 5 wherein:  
2 the watchdog timer devices issue a reset command if either of the watchdog timer devices  
3 do not receive a clear timer command from the watchdog drivers in a predetermined period of  
4 time.

1 7. An application watchdog, comprising a dedicated watchdog counter in the hardware layer  
2 of a computer system, and a watchdog driver operating in the kernel mode of the computer  
3 operating system, the watchdog driver comprising:

4 a system thread configured to monitor a plurality of designated user  
5 applications operating in the user mode of the computer operating system;

6 a message passing interface for receiving periodic signals from each of the  
7 user applications; and

8 a communication interface for transmitting a timer reset command to the  
9 dedicated watchdog counter;

10 wherein if the system thread receives a message from each of the designated user  
11 applications within an allotted period of time, the watchdog driver sends a timer reset command to  
12 the dedicated watchdog counter and wherein if the system thread does not receive a message from  
13 each of the designated user applications within the allotted period of time, the watchdog driver  
14 does not send a timer reset command to the dedicated watchdog counter.

1 8. The application watchdog of claim 7 wherein:

2 if the watchdog counter does receive a timer reset command from the watchdog driver, the  
3 counter is reset to begin counting down from the maximum allotted period of time and wherein if  
4 the watchdog counter does not receive a timer reset command from the watchdog driver, the  
5 counter is configured to restart the computer system when the counter expires.

6 9. The application watchdog of claim 8 wherein the watchdog counter further comprises:  
7 a timer value register that stores a digital representation of the maximum allotted period of  
8 time; and

9 a control and status register that comprises:

10 a bit for enabling the application watchdog;

11 a bit for counter reset;

12 bit fields for enabling early expiration warnings; and

13 bit fields for early expiration warning signals;

9 wherein if the watchdog counter does not receive a timer reset command from the  
10 watchdog driver and the early expiration warnings are enabled, the counter is configured to  
11 transmit early expiration warnings to the rest of the computer system before the counter expires.

1 10. The application watchdog of claim 9 wherein:  
2 the early warning messages are non-mask interrupts.

1 11. The application watchdog of claim 9 wherein:  
2 the early warning messages are maskable interrupts.

1 12. The application watchdog of claim 9 wherein:  
2 the early warning messages are system management interrupts.

1 13. The application watchdog of claim 7 wherein:  
2 the messages from the designated user applications are sent periodically by the applications  
3 and directed specifically to the watchdog driver.

1 14. The application watchdog of claim 7 wherein:  
2 the plurality of the user applications are prioritized by a computer user to permit varying  
3 levels of watchdog protection.

1 15. The application watchdog of claim 7 wherein:

the application watchdog operates in conjunction with a system watchdog that is configured to monitor the computer operating system for periodic activity; and

wherein both the application watchdog and the system watchdog are sufficiently configured to restart the computer system if either watchdog does not receive a timer reset command within an allotted period of time.

16. A method of detecting and restarting an unresponsive computer application, comprising:

- executing the application in a first protective layer of a computer operating system;
- executing an application watchdog driver in a second, more protected, protective layer of the computer operating system;
- establishing a message passing interface between the application and the watchdog driver;
- periodically transmitting signals from the application to the message passing interface;
- executing a system thread in the watchdog driver that is configured to monitor the message passing interface for the periodic signals from said application or other designated applications;

and

- using a dedicated watchdog timer device to count from a programmable initial value to a final system reset value;

wherein if the system thread detects a periodic signal from the application before the watchdog timer counts to the final system reset value, the watchdog driver initiates a command to the watchdog timer to reset the watchdog timer to the initial value and wherein if the system thread fails to detect a periodic signal from the application before the watchdog timer counts to the final system reset value, the watchdog timer initiates a command to restart the computer system.

1 17. The method of claim 16 further comprising:

2 sending an early warning message to notify system management software or firmware that  
3 the watchdog timer is about to expire.

1 18. The method of claim 16 wherein the initialization of the watchdog driver comprises:

2 loading the watchdog driver as the operating system loads following a computer system  
3 boot; and

4 loading and creating an initial input/output control signal interface that establishes the  
5 message passing interface.

1 19. The method of claim 18 wherein the initialization of the computer application comprises:

2 linking the application with a dynamic link library;

3 calling the watchdog driver via the dynamic link library and through the initial input/output  
4 control signal interface to validate the message passing interface; and

5 sending application location and identification information to the watchdog driver.

1 20. The method of claim 19 wherein the initialization of the watchdog timer device comprises:

2 setting the timer initialization value in a timer value register in the watchdog timer device;

3 and

4 setting the counter enable bit and early warning enable bits in a control/status register in the  
5 watchdog timer device.

1 21. The method of claim 17 wherein:

2 the early warning messages are NMI and SMI interrupts that are sent 9 seconds before the  
3 watchdog timer device expires.

1 22. The method of claim 16 wherein:

2 the system thread must detect a periodic signal from all designated applications before  
3 initiating the command to the watchdog timer to reset the watchdog timer to the initial value.

1 23. A computer system, comprising:

2 an operating system with at least two protection layers;

3 one or more computer applications; and

4 at least two watchdog drivers;

5 wherein a first of the plurality of watchdog drivers is configured to monitor the operating system  
6 for periodic messages and a second of the plurality of watchdog drivers is configured to monitor  
7 the computer applications for periodic messages; and

8 wherein if the second watchdog driver receives a periodic message from the computer applications  
9 in a predetermined period of time, the second watchdog driver delivers a command to clear the  
10 second of the plurality of watchdog timer devices.

1 24. The computer system of claim 23 wherein:

2 if the first watchdog driver receives a periodic message from the operating system in a  
3 predetermined period of time, the first watchdog driver delivers a command to clear the first of the  
4 plurality of watchdog timer devices.

1 25. The computer system of claim 24 wherein:  
2 the watchdog timer devices are configured to restart the computer system if either of the  
3 watchdog timer devices do not receive a clear timer command from the watchdog drivers in a  
4 predetermined period of time.

1 26. The computer system of claim 23 wherein:  
2 the watchdog driver creates timer events in the operating system scheduler that alert the  
3 watchdog driver when the predetermined period of time has expired.

1 27. A computer server, comprising:  
2 a central processing unit ("CPU") configured to execute an operating system and key,  
3 designated user applications;  
4 a system memory coupled to said CPU;  
5 an input/output processor ("IOP") configured to control server management architecture;  
6 a system watchdog device configured to receive periodic messages from the operating  
7 system; and

8 an application watchdog device configured to receive periodic messages from the user  
9 applications;

10 wherein if either the system watchdog device or the application watchdog device does not receive a  
11 periodic message for a designated period of time, the watchdog device that does not receive the  
12 periodic messages initiates a command to the CPU to reset the server.

1 28. The computer server of claim 27 wherein:



2 the system watchdog and application watchdog may be selectably enabled or disabled  
3 independent of one another.

1 29. The computer server of claim 28 wherein:  
2 the watchdog devices are selectably configured to transmit an early warning interrupt to the  
3 CPU before the watchdog device initiates the server reset command.

1 30. The computer server of claim 28 wherein:  
2 the watchdog devices are selectably configured to transmit an early warning notification to  
3 the IOP before the watchdog device initiates the server reset command.

1 31. The computer server of claim 28 wherein:  
2 the watchdog devices are selectably configured to transmit an event notification to the IOP  
3 when the watchdog device initiates the server reset command.